Motion Pattern Interpretation and Detection for Tracking Moving Vehicles in Airborne Video

Qian Yu and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089
qianyu@usc.edu, medioni@usc.edu

Abstract

Detection and tracking of moving vehicles in airborne videos is a challenging problem. Many approaches have been proposed to improve motion segmentation on frame-by-frame and pixel-by-pixel bases, however, little attention has been paid to analyze the long-term motion pattern, which is a distinctive property for moving vehicles in airborne videos. In this paper, we provide a straightforward geometric interpretation of a general motion pattern in 4D space (x, y, v_x, v_y). We propose to use the Tensor Voting computational framework to detect and segment such motion patterns in 4D space. Specifically, in airborne videos, we analyze the essential difference in motion patterns caused by parallax and independent moving objects, which leads to a practical method for segmenting motion patterns (flows) created by moving vehicles in stabilized airborne videos. The flows are used in turn to facilitate detection and tracking of each individual object in the flow. Conceptually, this approach is similar to "track-before-detect" techniques, which involves temporal information in the process as early as possible. As shown in the experiments, many difficult cases in airborne videos, such as parallax, noisy background modeling and long term occlusions, can be addressed by our approach.

1. Introduction

Detecting and tracking multiple moving vehicles from an airborne camera is a challenging problem and has drawn significant attention. As the size of vehicles is relatively small from an airborne view, appearance based detectors suffer from lack of resolution and blurry images. The motion-based detection approach relies on the stabilization of the camera motion using parametric models. Moving objects are defined as the areas that have not been stabilized. This method works well when the scene can be considered planar, or when the motion of the camera is pan/tilt/zoom. Otherwise, 3D depth in the scene produces pixel displacement, which cannot be accounted for by the global parametric model, usually termed as parallax. Figure 1 shows one example of noisy motion detection caused by parallax, which severely affects object detection and tracking. Besides parallax, many other cases affect detection and tracking in airborne videos, such as abrupt illumination changes, registration errors and occlusions.

(a) (b)

Figure 1. One parallax example (a) a typical scenario from a UAV camera (b) the residual image after subtracting background

Many approaches have been proposed to improve motion detection and tracking on frame-by-frame and pixel-by-pixel bases, *e.g*. global illumination compensation [9], parallax filtering [10], or detection using contextual information [4, 11]. No much attention has been paid on analyzing the long-term motion pattern of moving objects, which is a distinctive property for moving vehicles in airborne videos. Conceptually similar to "track-before-detect" techniques, we aim to involve temporal information in process as early as possible. Indeed, detection and tracking are coupled: if perfect detection is given, tracking becomes relatively straightforward, on the other hand, if we know the motion and trajectory of an object, detection is easier. Thus, in our approach, we first analyze the motion pattern of objects over a long time period before segmenting individual objects, and use the motion pattern in turn to facilitate the detection and tracking in each frame. By posing the detection and tracking task in this coupled framework, we can

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

deal with many difficult cases that challenge existing detection and tracking methods. Moreover, we do not assume any particular motion model of the moving objects, but simply assume objects are moving in a smooth way, which is a reasonable assumption for moving vehicles in airborne videos.

To analyze motion pattern, we first provide a clear definition in a 4D space, (x, y, v_x, v_y), and we further provide a geometric interpretation of a motion pattern in this 4D space. According to this geometric property, we propose to use Tensor Voting [12] to detect and segment motion patterns for general objects from noisy input data. Specifically, we analyze the motion patterns in airborne videos. Furthermore, we propose to use a two-step voting to segment motion patterns (flows) created by moving vehicles. The idea of defining voting strength in Tensor Voting is used to stitch fragmented flows caused by occlusions. Finally, segmentation and tracking of each object is performed in each flow with local kinematics and environmental constraints (entries and exits).

The key contributions of our approach are as follows. First, we provide a straightforward geometric interpretation of motion pattern in the 4D space and propose to use the Tensor Voting computational framework to detect and segment motion patterns. Second, we propose a practical method to use general motion pattern to facilitate segmentation, tracking and reacquisition of moving vehicles in airborne videos. This method solves many cases that are difficult to solve on a frame-by-frame basis.

The rest of this paper is organized as follows. In Section 2, we discuss related work in motion pattern analysis and tracking in airborne videos. In Section 3, we present the overview of our approach. In Section 4, we present the general motion pattern detection and a special case for airborne videos. Section 5 introduces detection and tracking of objects in motion patterns. Experimental results are shown in Section 6 followed by conclusions at the end.

2. Related work

Detecting and tracking multiple moving vehicles from an airborne camera is a challenging problem. Some existing methods and systems [13, 1] have demonstrated good results in planar (or quasi-planar) scenes where good motion segmentation results can be achieved. A scene that contains strong parallax is still difficult for existing methods. Some work has been proposed to deal with the parallax problem. One way is to explicitly recover the depth of the scene, and then to remove everything far above the ground plane [11]. Another approach to remove parallax is to characterize the parallax and motion by using geometry constraints as in [10], where epipolar and relative depth constraints are applied to filter parallax without explicitly estimating the depth. This approach may fail on some degenerate cases. More importantly, parallax filtering in a frame-by-frame and pixel-by-pixel manner is very noisy [10].

Contextual information has been applied to improve detection, tracking and reacquisition, as in [11, 14, 15]. In [14], a color-based scene segmentation is used to determine the regions of foliage, grass and road *etc*., thus to remove false alarms in motion detection. In [11], geo-registration, depth map and GIS information with road network are used to remove false alarms out of road. In our approach, the motion patterns created by vehicles are flows, which are similar to the road concept but work for unstructured environment. In [15], Ali *et al*. originally proposed to use objects that are moving in a similar context to predict and reacquire objects after long term occlusion. This method assumes that low level motion detection and tracking have been solved and the reacquisition is performed at the object level. The errors in low level motion segmentation under strong parallax situation are not considered in this method. In our approach, we aim to use the context information to improve both low level motion segmentation and high level reacquisition even when there is only one single object in the scene.

Motion pattern analysis before tracking each object has started to get attention in recent years, especially for crowded scenarios [3, 5, 16, 17], where tracking each individual is very difficult. In [16], Lagrangian Particle Dynamics is used to segment high density crowd flows and further track each marked objects as in [3]. In [5, 17], a clustering based method is proposed to segment and represent the dense motion flow in crowded scenes. These methods all apply pre-filtering or pre-clustering steps (median filters in [16], Gaussian ART in [17]) remove the outliers. As the scene may contain different motion patterns at one location within a period of time, (such as at a road intersection), averaging or filtering before knowing the local structure of motion patterns may destroy such structure. Our method analyzes motion patterns and filters out noise in input data under a unified computational framework.

Our work is also related to Min's work in [6], which aims to segment each motion region using Tensor Voting in a 5D space (x, y, v_x, v_y, t). However, the goal to segment each moving region is too difficult to achieve due to the lack of samples in a large voting space. In our approach, since the time dimension collapses, objects moving in a similar manner at different times share the same motion pattern, and thus reinforce the motion pattern in the 4D space.

3. Overview

The pipeline of our detection and tracking approach contains three phases, shown in Figure 2. In the first phase, affine motion compensation and detection are applied to estimate the transformation between consecutive frames and model dynamic background for each frame. Residual pixels after motion compensation account for independent moving objects, noise in motion compensation, or parallax. Before

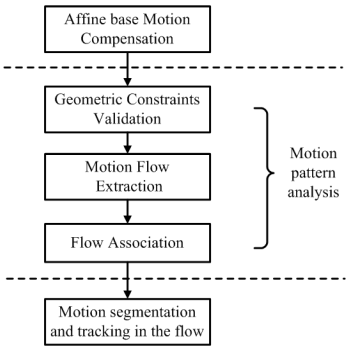


Figure 2. Overview of the proposed approach

segmenting moving objects from a residual image at each frame, in the second phase, we analyze the general motion pattern created by moving vehicles and segment their motion patterns (flows) over a long period of time. In the third phase, we detect moving objects in the flow and associate them according to flow dynamics and appearance similarity. Following the concept of "track-before-detect", our approach involves temporal information as early as possible, and in return uses the motion pattern to improve detection and tracking in each frame.

The affine motion detection framework [7] initially extracts a number of feature points in each frame. Then the feature points in consecutive frames I_t and $I_t + 1$ are matched by evaluating the cross-correlation of local windows around feature points. A 2D affine motion model $A_{t+1,t}$ is robustly estimated by a RANSAC-based scheme. The affine motion model $A_{t+1,t}$ globally compensates for the motion from I_t to I_{t+1}. This affine model is not only used for motion compensation and detection, but also to warp motion vectors from different frames to a mosaic space for motion pattern analysis. The pixels that do not satisfy this motion model are classified as residual pixels.

In order to detect and segment the motion flow caused by moving vehicles, we first compute optical flows on all the residual pixels between each frame and its stabilized next frame. We further warp these optical flows into a mosaic space, *i.e*. common reference coordinates, where the camera motion is compensated. Then, we cast the warped optical flows into a 4D space, and use Tensor Voting to analyze their geometric property and filter out parallax and noise. For each flow, we use a meanshift based method [3] to find its endpoints (entry and exit). Due to long term occlusions, the detected flow may be fragmented. We employ the Hungarian [2] algorithm to link the endpoints of the flows according to motion smoothness.

Given the flow information, we compute the local dynamics at each location in the flow. To segment each object, we first use the motion history image (MHI) method [13] to generate an initial segmentation, and then associate segmented regions according to their appearance similarity and flow dynamics to generate tracklets. The Hungarian algorithm is applied again to associate tracklets to form consistent long tracks. The end (entry and exit) information of a flow is imposed as environmental constraints when associating tracklets.

4. Motion Pattern Analysis

In this section, we first address the general motion pattern analysis, and then discuss the specific property of the motion pattern created by moving vehicles in airborne videos.

4.1. Motion Flow and Pattern

Consider a 2D point P smoothly traversing in a spatio-temporal space. By projecting the motion of the point in a 4D space, (x, y, v_x, v_y), where (x, y) is the location of the point and (v_x, v_y) denotes the time derivatives of motion along x and y axes, we obtain a *fiber* (dimensionality is one) in the 4D space that represents the motion characteristic of that point. If a set of 2D points (*e.g*. on the same object) are moving in a similar way, a bundle of fibers form a *flow*. Many types of object motions can be represented by a flow. A single moving vehicle or a convoy of vehicles observed from an airborne camera are such typical cases.

Generally, we define the motion pattern in the 4D space as a set of motion vectors,

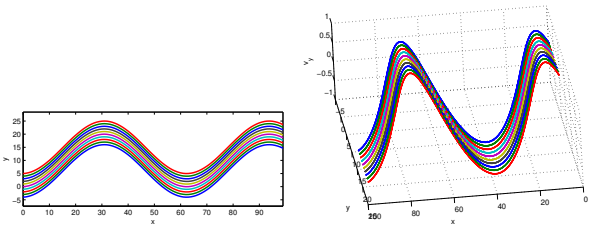
$$\mathcal{F} = \{(x, y, v_x, v_y), (x, y) \in R^2\}. \quad (1)$$

Without loss of generality, in one motion pattern, one motion vector (v_x, v_y) is assigned at one location (x, y), *i.e*. (v_x, v_y) is a function of (x, y), $(v_x, v_y) = \mathcal{F}(x, y)$. Note that there may exist multiple motion patterns at the same location, (*e.g*. at a road intersection). Objects whose motion complies with the same motion pattern are called objects moving in the same *motion context*. A motion flow can be regarded as one particular type of motion pattern.

The motion pattern defined in Eq.1 essentially describes the general motion characteristics of objects over a period of time. In practice, the motion estimation of one object at a time inevitably contains noise. The estimated motion vectors in a motion pattern $\mathcal{F}$ can be written as $f = (x, y, \mathcal{F}(x, y) + e)$, where e accounts for the noise in motion estimation. We aim to analyze the general motion pattern from multiple noisy motion vectors over time, and then use this information to facilitate detection and tracking of each object in the motion pattern.

The essential property of a motion pattern is that *each smooth motion pattern corresponds to a smooth sheet in the 4D space*, *i.e*. the local dimensionality is 2. It is easy to know that the dimensionality is 2, since

- the projection of a motion pattern in (x, y) space is 2 in non-degenerate cases, thus the dimensionality in the 4D space is no less than two;



(a) a motion flow generated with $v_x = 1, v_y = \sin(0.1t)$ (b) the corresponding structure in (x, y, v_y) while v_x is a constant

Figure 3. Illustration of a motion flow, which has a sheet property in 4D space

- at most one motion vector is assigned at one location, thus the local dimensionality is no more than two.

According to the smooth motion assumption, the normal on the sheet created by one motion pattern changes smoothly and different motion patterns produce discontinuity between them in 4D space. Noise caused by erroneous optical flows do not form a coherent sheet with local smoothness. One example of a simulated flow is shown in Figure 3.

4.2. Tensor Voting and Motion Pattern Segmentation

Suppose we have a set of noisy input samples in the 4D space, structures we aim to find in this space are smooth 2D sheets. We analyze vectors that span the normal and tangent space at each point to infer the geometric structure while filtering noise out, and use local smoothness to segment one structure from others. We use Tensor Voting to achieve this task.

Tensor Voting [12] can be regarded as an unsupervised computational framework to estimate geometric information. Tensor voting has been proved capable of estimating structures in N-D space with very noisy input data. In Tensor Voting framework, the local geometric information at one point in N-D space is encoded in a symmetric, nonnegative definite matrix. The local geometry can be derived by examining the eigensystem. Recall that a tensor can be decomposed as

$$T = \sum_{i=1}^{N} \lambda_i e_i e_i^T = \sum_{i=1}^{N-1} (\lambda_i - \lambda_{i+1}) \sum_{k=1}^{i} e_k e_k^T + \lambda_N \sum_{k=1}^{N} e_i e_i^T \quad (2)$$

where $\{\lambda_i\}$ are the eigenvalues arranged in descending order, $\{e_i\}$ are the corresponding eigenvectors, and N is the dimensionality of the input space. The decomposition in Eq.2 provides a way to interpret the local geometry. The largest gap between two consecutive eigenvalues, $\lambda_i - \lambda_{i+1}$, indicates the dimensionality d,

$$d = \arg\max_i (\lambda_i - \lambda_{i+1}) \quad (3)$$

The largest difference value $\lambda_d - \lambda_{d+1}$ is the saliency of the dimensionality. In order words, a geometric structure, whose normal space is d-dimension and the tangent space is $(N-d)$-dimension, is the most salient interpretation according to T. The corresponding eigenvectors $\{e_1, ..., e_d\}$ span the normal space of the structure and $e_{d+1}, ..., e_N$ span the tangent space. In our case, we are interested in the structures whose normal space's dimensionality is 2 in the 4D space.

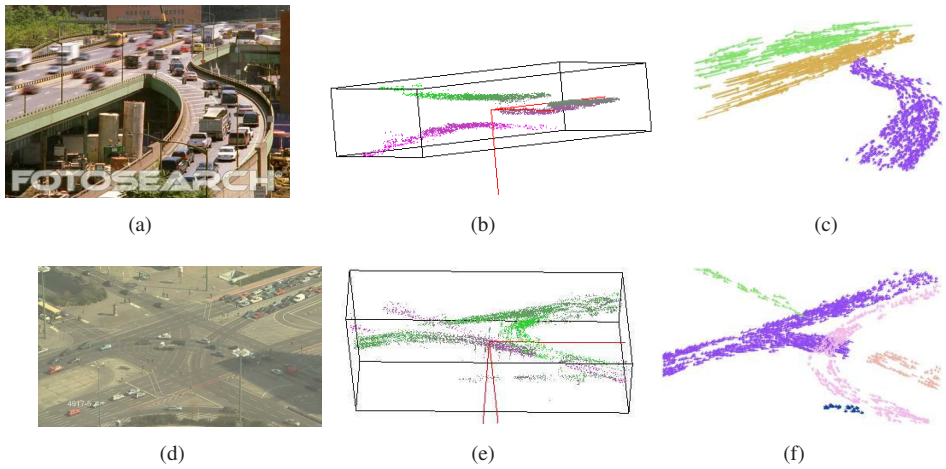
Given the input data, a set of 4D motion vectors, $\{f_i\}$, we encode each sample as a ball tensor, which indicates no orientation, since at the beginning we have no knowledge of the local structure at a point. Each f_i receives a vote $T_{j \to i}$ from its neighbors f_j in 4D space. The voting result at one point, which indicates its geometric property, is obtained by adding up all the incoming votes from its neighbors. The vote from a voter f_i to a receiver f_j encodes the tensor at f_i, the orientation and the distance from f_i to f_j. The result of this process can be interpreted as a local, nonparametric estimation of the geometric structure at each sample position. After accumulating all cast tensors, the local geometry can be interpreted according to of Eq.3.

In our motion pattern segmentation, we take original optical flows computed in multiple frames as input, without any pruning or clustering. After the voting process, we examine the cast tensor T and keep the structures of dimensionality 2 with saliency larger than a threshold. After this tensor voting process, most of the structures created by noise are filtered out. Since, at one 2D location (x, y), there may exist multiple motion vectors that belong to the same motion pattern, we use the average of the motion vectors to represent the estimated motion pattern. Note that, we only average motion vectors on the same sheet in the 4D space. This averaging is essentially different with prefiltering, since it is performed after we have the knowledge of local structures and filter noise out.

After detecting the desired structure and filtering out the noise, we use a flood-fill algorithm in 4D space to segment each motion pattern. According to the smooth motion assumption, the sheet formed by one motion context has local smoothness and discontinuity exists between sheets caused by different motion patterns. The neighbor samples in 4D space that have similar normal are assigned the same label. We use principle angles [8] to measure the similarity between two normal spaces. Two examples of motion pattern are shown in Figure 4. The video used in the second example is at a road intersection, where exist multiple motion patterns at some location. Directly smoothing in the (x, y) space will destroy such motion patterns.

4.3. Motion Pattern Analysis for Airborne Videos

We have discussed the properties of general motion patterns in 4D space. Now, we analyze motion patterns in airborne videos, where we aim to find those created by moving vehicles. The pixels that do not satisfy the global motion



(a) (b) (c)

(d) (e) (f)

Figure 4. One example of motion pattern segmentation (a)(d) scenes with traffic flows (b)(e) motion pattern shown in (x, y, v_x) space (c)(f) segmentation of motion patterns

model are classified as residual pixels. The residual pixels account for noisy background modeling, independent motion, and parallax. We compute optical flows on the residual pixels between the frame I_t and its stabilized next frame $A_{t+1,t}I_{t+1}$. Then, the optical flow is warped to a common reference frame, which we call the mosaic space. We select the first frame as the reference frame. If geo-registration is available, the geo-map can also be selected as a mosaic space. The global camera motion is compensated in the mosaic space, where motion pattern analysis is performed. Residual pixels caused by noise do not form a consistent motion pattern, thus can be filtered by examining their dimensionality as discussed in Section 4.2.

The essential difference between motion patterns created by parallax and by moving vehicles is as follows. After principal motion estimation, the motion pattern of moving vehicles is generated by the intrinsic properties of the objects and static environmental constraints on the ground plane (*e.g.* road, or non-road area), which is independent of camera motion. The motion pattern of parallax is caused by the camera motion, as each motion vector on a 3D structure should be along the epipolar line that is determined by camera's translation. According to the relative affine structure [18], the projection p_t of a 3D point P on I_t can be decomposed as,

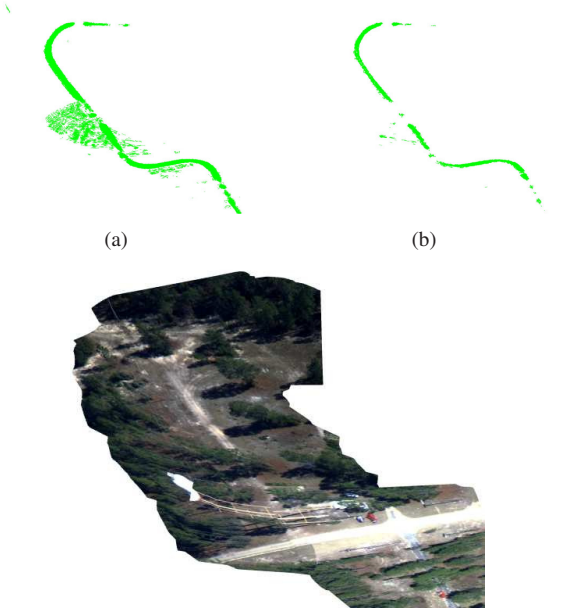
$$p_t = \underbrace{A_{t,r}p_r}_{(1)} + \underbrace{ke_t}_{(2)} \tag{4}$$

where p_r is the projection of P in the reference frame, k is a scalar, which is independent of the camera pose at time t, and e_t is the epipole at time t. The first term in Eq.4 is compensated for by the affine motion. From the second term, we can see the motion of parallax (ke_t) is indeed determined by the camera motion. Interestingly, when the epipole is moving in a non-smooth way, the motion of parallax cannot form smooth patterns, thus non-smooth epipole motion actually helps us to remove parallax.

When the camera is moving in a smooth way, however, the parallax can still form a smooth motion pattern. In other words, it also exhibits as 2D structures in the 4D space. Specifically, in airborne videos, the motion patterns of moving vehicles forms *flows*. Such flow shows a fiber property (dimensionality is 1) on a larger scale. This property is due to the fact that the motion range of a vehicle over time is much larger than its 2D dimensions. This is derived from the characteristics of moving vehicles in airborne videos. In order to examine the geometric property at a larger scale, we can simply enlarge the voting scale. In our experiments, we observe that, when we enlarge the scale of voting, the motion field caused by a small 3D structure becomes a point tensor or it remains a sheet for a large 3D structure. Thus, the procedure of segmenting the motion field created by moving vehicles is: first vote at a small scale and keep only the 2D structures to remove noise, and then vote at a large scale, keep only the 1D fiber structures. In practice, instead of directly enlarging the voting scale, we down-sample the 4D space to achieve an efficient implementation. We show one example of motion flow detection in an airborne video in Figure 5. There exists strong parallax (a water tower) in this video. After the first scale voting, some motion patterns created by the parallax still remains, shown in Figure 5(a). After changing the voting scale, the parallax motion pattern is filtered out, shown in Figure 5(b).

4.4. Stitching Flows

The motion flow created by moving vehicles may be fragmented due to occlusion. Thus, we propose a method to stitch them up. After we find each flow, we randomly place several "floats" (square 2D Gaussian kernels) in the flow and apply the meanshift like method used in [5, 3] along



(a) (b) (c)

Figure 5. Voting at two scales to detect motion patterns by moving vehicles (a) Tensor Voting at a small scale and keep sheets (b) second step voting at a larger scale and keep fibers (c) the mosaic image (space) used for warping optical flows

both positive and negative directions, the "floats" terminate at the ends of the flow. Then, we cluster termination points that are consistent with the ends found by shape analysis. Note that we use clustering only for obtaining a more accurate end location, but the clustering method [5] cannot be used to filter out motion field of parallax. After we have the ends (both entry and exit) of the flows, we use a vote-casting method inspired from Tensor Voting to calculate the motion consistency between flows as shown in Figure 6.

Let O denote one end of a flow, $\vec{N}$ denote its normal, which is known after we find the flow, and we want to compute its motion consistency with another end from a different flow P. The motion consistency should consider both orientation and strength. As can be seen in Figure6, the ideal orientation $\vec{N}_{O\to P}$ (gray arrow starting from P) is given by drawing a big circle whose center C is in the line of $\vec{N}$ and it passes both O and P while preserving the normal $\vec{N}$. The ideal orientation ensures the smoothest connection between two ends, O and P. The actual normal at

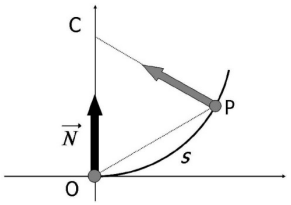


Figure 6. Computing motion consistency between two flows, derived from the concept of casting vote in tensor voting

P is $\vec{N}_P$. The consistency between O and P is computed by the following function:

$$S(O,P) = \exp(-|s|^2 - ck^2)(\vec{N}_{O\to P} \cdot \vec{N}_P) \qquad (5)$$

Here, $|s|$ is the arc length, k is the curvature, c is the decay rate. Note that besides the introduction of the dot product, the scale σ in Tensor Voting's decay function [12] is gone, since there is no concept of neighbors, *i.e.* any two ends from different flows within a gating threshold can be associated. After we calculate the motion consistency between each pair of flow ends, we use the Hungarian algorithm [2] to find the best associations.

5. Detection and Tracking in the Flow

Given the flow information, detection and tracking becomes much easier. First, most of the residual pixels caused by noise and parallax have been filtered out. Second, the local dynamics in the motion field are known. Third, the entries and exits of flows, which actually reflects the environmental information, can be imposed during the data association: both termination and birth of a track are captured by this information. Note that one flow may contain multiple moving vehicles moving in sequence or in parallel, but their motion should comply with the motion pattern.

In residual images along the flows, we adopt the motion history image method proposed in [13] to segment independent motion regions. Each segmented region is represented as an oriented rectangle. An association score between regions R_i and R_j from neighboring ($|i-j| \le \delta$) frames encodes both appearance similarity and consistency with the local motion field as:

$$p_{ij} = CS_{ij}e^{-\frac{|R(i)-\bar{R}_j(i)|+|R(j)-\bar{R}_i(j)|}{2}} \qquad (6)$$

The appearance similarity S_{ij} is simply the normalized correlation between two image patches, $\bar{R}_j(i)$ is the predicted location from j to i by using $|i-j|$ steps of mean shift (the direction is $sign(i-j)$) in the motion field. According to this similarity measure, we aggregate these isolated regions from different frames into tracklets. We further filter out isolated regions and very short tracklets that come from noisy motion segmentation. For a pre-filtered tracklet, we use the average image patch of the oriented rectangles as its appearance template. A local translation relaxing is used to find the best matching location for averaging appearance template. The motion of a tracklet is encoded in its start and end points. The motion consistency between tracklets (a start with a end) is computed as in Eq.5. Then, we apply the Hungarian algorithm to associate tracklets into tracks. Here, we encode the entry and exit information of a flow in the utility matrix used in the Hungarian algorithm. Suppose there are n tracklets in the

pool, the utility matrix $\mathcal{A}_{2n\times 2n}$ is a matrix of size $2n \times 2n$. $\mathcal{A}_{(1,\cdots,n)\times(1,\cdots,n)}$, except its diagonal elements, contains the similarity between any pair of tracklets, the diagonal of $\mathcal{A}_{(n+1,\cdots,2n)\times(1,\cdots,n)}$ stores the termination probability of each tracklet, which is computed according to the distance between the end point of a tracklet and the exit of the flow; the diagonal of $\mathcal{A}_{(1,\cdots,n)\times(n+1,\cdots,2n)}$ stores the birth probability of each tracklet, which is computed according to the distance between the start point of a tracklet and the entry of the flow. All the other elements in $\mathcal{A}$ are zero. By expanding the similarity matrix, we impose the environmental information in the tracklet association to avoid the fragmented tracks in the middle of a flow. Note that the tracklet association is performed among flows that have been stitched in the motion pattern analysis phase.

6. Experimental Results

Results on two videos (around 1800 frames for each video) are shown in this section, we have submitted supplementary materials for more visual results. In these two videos, we show that some difficult problems (including parallax, noise in background modeling and long term occlusions), that challenge the existing UAV tracking systems, can be addressed by our approach. We also demonstrate our approach can reacquire objects across occlusions and can maintain track identifications for the case of leaving and reentering the field of view (FOV) of the camera. Our approach combines segmentation, tracking and reacquisition in a unified framework. For all the experiments, we use the KLT optical flow method. Optical flows are computed only at the residual pixels. To segment the flow, we need to provide two scales in Tensor Voting for detecting object motion pattern. Those are automatically determined by the number of neighbors that one sample should have at each scale in 4D space.

In the first video shown in Figure 7, moving vehicles often leave and re-enter the FOV of the camera. Also, the vehicles in this video undergo a "U-turn" motion, which is hard to describe with a parametric motion model. After computing the flow dynamics and stitching the flows, then identifications of objects that fall out of the FOV are maintained consistently in the whole video. The second video shown in Figure 8, contains strong parallax and a convoy of vehicles passing through a forrest where long term occlusions occur. This video challenges existing motion segmentation and tracking methods. The residual pixels that do not belong to valid motion patterns are shown in red in Figure 8. Such regions caused by parallax, which sometimes form larger regions than moving objects, cannot be filtered out by morphological operations or the motion history image method. By flow stitching, the long occlusion is correctly handled in the forest video. In both Figure 7 and Figure 8, we show the estimated the motion field after flow segmentation in both the mosaic space and the image space.

7. Summary

We have proposed an approach to detect and segment general motion patterns (for static cameras, or moving cameras) with noisy input data by using Tensor Voting in the 4D space (x, y, v_x, v_y). An straightforward geometric interpretation of a motion pattern in this 4D space is provided. Also, we presented a method to segment flows created by moving vehicles in airborne videos and in turn facilitate the detection and tracking of each object in the flow.

Our framework has no confliction with parallax filtering method using geometric constraints. A combined solution can be achieved easily. Our method currently uses a relatively long sequence to detect motion patterns. We will apply sliding window techniques to leverage the delay issue. In the future, we will investigate video retrieval and abnormal event analysis using this motion pattern analysis instead of tracking each individual object.

Acknowledgements

This work was supported by MURI-ARO W911NF-06-1-0094. The authors would like to thank Dr. Chang Huang and Dr. Weikai Liao for their comments and discussion.

References

[1] S. Ali and M. Shah. Cocoa - tracking in aerial imagery. In *SPIE*, 2006.

[2] H. W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, pages 83–97, 1955.

[3] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *ECCV*, 2008.

[4] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[5] M. Hu, S. Ali, and M. Shah. Detecting global motion patterns in complex videos. In *ICPR*, 2008.

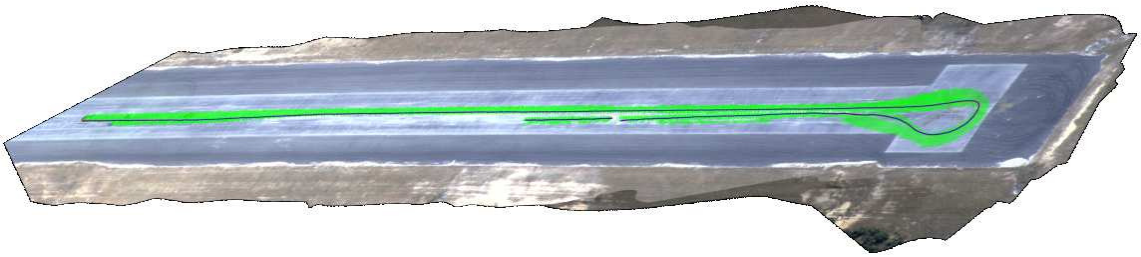
[6] C. Min and G. Medioni. Inferring segmented dense motion layers using 5d tensor voting. *PAMI*, 30(9):1589–1602, 2008.

[7] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *CVPR*, pages 267–272, 2003.

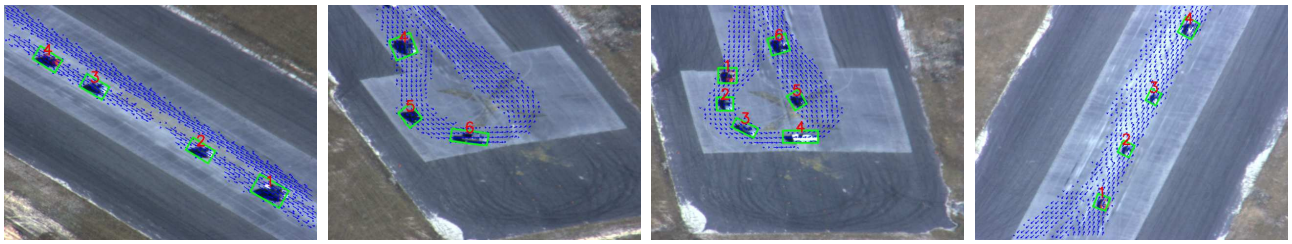
[8] A. Bjoerck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, pages 579–594, 1973.

[9] H. Yalcin, R. Collins, and M. Hebert. Background estimation under rapid gain change in thermal imagery. In *Object Tracking and Classification in and Beyond the Visible Spectrum*, 2005.

[10] C. Yuan, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in presence of strong parallax from a moving camera by multi-view geometric constraints. In *PAMI*, volume 29, pages 1627–1641, 2007.



(a) The motion field with ends and entry-to-sink paths



(b) Snapshots of tracking multiple vehicles in the flows. Track identification is maintained throughout the video.

Figure 7. Tracking with maneuvering objects

(a) a close look of motion field with ends and entry-to-sink paths

(b) Snapshots of tracking multiple vehicles in the flows. Red indicates the residual pixels that do not belong to valid motion patterns.

Figure 8. Tracking with strong parallax

[11] J. Xiao, H. Cheng, F. Han, and H. Sawhney. Geo-spatial aerial video processing for scene understanding. In *CVPR*, pages 1–8, 2008.

[12] C. keung Tang, G. Medioni, and M. suen Lee. N-dimensional tensor voting, application to epipolar geometry estimation. *IEEE PAMI*, pages 829–844, 2001.

[13] Z. Yin and R. T. Collins. Moving object localization in thermal imagery by forward-backward mhi. In *CVPR Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS)*, 2006.

[14] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *CVPR*, pages 990–997, 2005.

[15] S. Ali, V. Reilly, and M. Shah. Motion and appearance contexts for tracking and re-acquiring targets in aerial videos. In *CVPR*, 2007.

[16] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, pages 1–8, 2007.

[17] M. Hu, S. Ali, and M. Shah. Learning motion patterns in crowded scenes using motion flow field. In *ICPR*, 2008.

[18] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3d reconstruction from perspective views. In *CVPR*, pages 483–489, 2004.